METHOD AND MECHANISM FOR SUPERPOSITIONING STATE VECTORS IN A SEMANTIC ABSTRACT

RELATED APPLICATION DATA

This application is a continuation-in-part of co-pending U.S. Patent Application Serial No. _____, titled "A METHOD AND MECHANISM FOR THE CREATION, MAINTENANCE, AND COMPARISON OF SEMANTIC ABSTRACTS," filed July 13, 2000, and is related to U.S. Patent No. 6,108,619, titled "METHOD AND APPARATUS FOR SEMANTIC CHARACTERIZATION," issued August 22, 2000, and to co-pending U.S. Patent application Serial No. 09/512,963, titled "CONSTRUCTION, MANIPULATION, AND COMPARISON OF A MULTI-DIMENSIONAL SEMANTIC SPACE," filed February 25, 2000, all commonly assigned.

FIELD OF THE INVENTION

This invention pertains to determining the semantic content of documents via computer, and more particularly to comparing the semantic content of documents to determine similarity.

BACKGROUND OF THE INVENTION

U.S. Patent Application Serial No. _____, titled "A METHOD AND MECHANISM FOR THE CREATION, MAINTENANCE, AND COMPARISON OF SEMANTIC ABSTRACTS," filed July 13, 2000, referred to as "the Semantic Abstract application" and incorporated by reference herein, describes a method and apparatus for creating and using semantic abstracts for content streams and repositories. Semantic abstracts as described in the Semantic Abstracts application include a set of state vectors. Thus, storing the semantic abstract requires storing each vector in the set, taking up a lot of storage space. Further, measuring the distance between a semantic abstract and a summary of a document using the Hausdorff distance function, a complicated function, requires numerous calculations along the way to calculate a single distance.

The Semantic Abstract application discusses techniques for simplifying the semantic abstract (e.g., by generating a centroid vector). Such techniques have limitations, however; most notably that particular information can be lost.

Accordingly, a need remains for a way to construct a single vector that captures the meaning of a semantic context represented by a clump of vectors without losing any information about the semantic context.

SUMMARY OF THE INVENTION

The invention is a method and apparatus constructing a single vector representing a semantic abstract in a topological vector space for a semantic content of a document. The semantic content is constructed for the document on a computer system. From the semantic content, lexemes or lexeme phrases are identified. State vectors are constructed for the lexemes/lexeme phrases. The state vectors are superpositioned into a single vector, which forms the semantic abstract for the document.

The foregoing and other features, objects, and advantages of the invention will become more readily apparent from the following detailed description, which proceeds with reference to the accompanying drawings.

BRIEF DESCRIPTION OF THE DRAWINGS

FIG. 1 shows a computer system on which the invention can operate to construct a single vector semantic abstract.

FIG. 2 shows a computer system on which the invention can operate to search for documents with content similar to a given semantic abstract.

FIG. 3 shows a two-dimensional topological vector space in which state vectors are used to determine a semantic abstract for a document.

FIG. 4 shows a two-dimensional topological vector space in which semantic abstracts for three documents are compared.

FIG. 5 is a flowchart of a method to construct a single vector semantic abstract for a document in the system of FIG. 1.

FIG. 6 shows a two-dimensional topological vector space in which state vectors have been clumped.

DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENT

FIG. 1 shows a computer system 105 on which a method and apparatus for using a multi-dimensional semantic space can operate. Computer system 105 conventionally includes a computer 110, a monitor 115, a keyboard 120, and a mouse 125. Optional equipment not shown in FIG. 1 can include a printer and other input/output devices. Also not shown in FIG. 1 are the conventional internal components of computer system 105: e.g., a central processing unit, memory, file system, etc.

Computer system 105 further includes software 130. In FIG. 1, software 130 includes semantic content 135, state vector constructor 140, and superposition unit 145. State vector constructor 140 takes lexemes and lexeme phrases from semantic content 135 and constructs state vectors for the lexemes/lexeme phrases in a topological vector space. Superposition unit 145 takes the state vectors constructed by state vector constructor 140 and superpositions them into a single vector for the semantic abstract. In the preferred embodiment, superposition unit 145 includes vector algebra unit 150. Vector algebra unit 150 adds the state vectors together to construct the single vector for the semantic abstract.

Although the above description of software 130 creates a single vector from state vectors in the topological vector space, the state vectors can be divided into groups, or clumps. This produces a minimal set of state vectors, as opposed to a single vector, which avoids distant lexemes/lexeme phrases from being superpositioned and losing too much context.

In the preferred embodiment, clumps are located by performing vector quantization, which determines a distance between each pair of state vectors; vectors sufficiently close to each other can then be clumped together. For example, a vector can be determined to be in a clump if its distance is no greater than a threshold distance to any other vector in the clump. FIG. 6 shows a two-dimensional topological vector space in which state vectors have been clumped. In FIG. 6, state vectors 605 have been grouped into three clumps 610, 615, and 620. The state vectors in each of clumps 610, 615, and 620 can then be superpositioned as described below, and the resulting three vectors grouped into a semantic abstract. Note that, in FIG. 6, not every state vector is part of a clump. For example, state vector 625, although close to vectors in both of clumps 615 and 620, is not sufficiently close to all of the vectors in either clump, and is excluded from both. Similarly, vector 630 is too far from any clump to be included, and is excluded from all clumps.

A person skilled in the art will recognize that other techniques can be used to locate clumps: for example, by dividing the vectors in groups so that each vector in a particular group has an angle within a certain range. The remainder of this invention description assumes that the state vectors form only a single clump and vector quantization is not required; a person skilled in the art will recognize how the invention can be modified when vector quantization is used.

Although the document from which semantic content 135 is determined can be found stored on computer system 105, this is not required. FIG. 1 shows computer system 105 accessing document 160 over network connection 165. Network connection 165 can include any kind of network connection. For example, network connection 165 can enable computer system 105 to access document 160 over a local area network (LAN), a wide area network (WAN), a global internetwork, a wireless network or broadcast network, or any other type of network. Similarly, once collected, the semantic abstract can be stored somewhere on computer system 105, or can be stored elsewhere using network connection 165.

FIG. 2 shows computer system 105 programmed with a different software package. In FIG. 2, computer system 105 includes software 205 to use semantic abstract 210 to find documents with similar content. Search means 215 searches the topological vector space for documents with semantic abstracts that are "similar" to semantic abstract 210. In the preferred embodiment, search means 215 is implemented as software to query the vector space for semantic abstracts close to the single vector in semantic abstract 210. What semantic abstracts qualify as "similar" to a given semantic abstract will be revisited with reference to FIG. 4 below. Retrieval means 220 retrieves the documents with semantic abstracts in the topological vector space similar to semantic abstract 210.

On the Meaning of the Meaning of Meaning

Recall the definition of a vector space. A nonempty set V is said to be a *vector space* over a field F if V is an abelian group under an operation denoted by +, and if for every $\alpha,\beta \in F$, $v,w \in V$ the following are satisfied:

- $\alpha(v + w) = \alpha v + \alpha w$
- $(\alpha + \beta)v = \alpha v + \beta v$
- $\alpha(\beta v) = (\alpha\beta)v$
- $1v = v$

where "1" represents the unit element of F under multiplication.

As shown in co-pending U.S. Patent application Serial No. 09/512,963, titled "CONSTRUCTION, MANIPULATION, AND COMPARISON OF A MULTI-DIMENSIONAL SEMANTIC SPACE," filed February 25, 2000, a set S of lexemes can be represented as a vector space. This representation is accomplished by introducing a topology τ on S that is compatible with the sense of the lexemes, building a directed set from the lexemes, and then (given the separation axioms) showing an explicit one-to-one, continuous, open mapping ϑ from S to a subspace of the Hilbert coordinate space – a *de facto* vector space. This open mapping ϑ is continuous and open with respect to τ, of course.

How is ϑ expressed? By the coordinate functions g_k: $S \Rightarrow \mathbf{I}^1$. And how are the g_k defined? By applying Urysohn's lemma to the k^{th} chain of the directed set, where A = {S - root}, B = $\overline{n_m}$ (the closure of the minimal node of the chain), and the intermediate nodes of the chain take the role of the *separating* sets (used to define the function predicted by Urysohn's lemma) $C(r/2^n)$, $U(r/2^n)$. (Of course in practice the continuous functions g_k can only be approximated with step functions, the resolution being constrained by the chain length.) In other words, the k^{th} chain provides a natural mechanism for defining g_k. Or to put it another way the k^{th} chain *identifies* g_k.

As is well known, functions that are nicely behaved can form a vector space, and it so happens that step functions are very well behaved indeed. Consider the vector space Q spanned by the coordinate functions g_k, where $q \in Q$ is of the form $\Sigma\lambda_k g_k$, $\lambda_k \in \mathbb{R}$ (the real numbers). Define an inner product on Q, of the form $\langle q_1, q_2 \rangle = \int q_1 \bullet q_2$, where it is understood that we integrate over S in a topologically consistent manner.

Given an inner product space Q, Q is a function space. In fact, Q is **the** function space spanned by the functions g_k. The functions g_k are defined by their corresponding chains. In fact the k^{th} chain *uniquely* identifies g_k, so that $\{g_k\}$ is more than simply a spanning set; it is a **basis** of Q.

Having built the metric space Q in such a way as to entail the topology on S, the next step is to coherently leverage S into a metric space via Q's structure. With the two metrics (of S and Q) commensurable, the goal of quantifying the notion of near and far in S will be accomplished.

By definition, if V is a vector space then its *dual space* is Hom(V, F). Hom(V, F) is the set of all vector space homomorphisms of V into F, also known as the space of *linear functionals*. So, the dual of Q (i.e., Hom(Q, $\mathbb{R}$)) is a function space on a function space.

Now, consider that for any $s \in S$, the function ε_s associates the function g_k with an element of the real field: $\varepsilon_s(g_k) = g_k(s)$. A simple check shows linearity, i.e., $\varepsilon_s(g_k + g_n) = (g_k + g_n)(s) = g_k(s) + g_n(s) = \varepsilon_s(g_k) + \varepsilon_s(g_n)$. The reader can similarly verify scalar multiplication. So, what does this show? It shows that **every element of S corresponds to an element of the dual of Q**. The notations $\varepsilon_s(g_k)$ and $s(g_k)$ are used interchangeably.

Now the notion of the dual of Q is "properly restricted" (limited to a proper subspace) to those linear functionals in the span of S: $\Sigma\lambda_k s_k$, $\lambda_k \in \mathbb{R}$, where it is understood that $(\lambda_i s_i + \lambda_j s_j)g_k = \lambda_i s_i(g_k) + \lambda_j s_j(g_k)$. When properly restricted, it can be shown that Q and its dual are isomorphic. Indeed, for the finite dimensional case it is very easy to prove that a vector space and its dual are isomorphic. So the dimension of the dual space of Q – i.e., the dimension of the space spanned by S in its new role as a set of linear functionals – is equal to the dimension of Q. And what does the linear functional s "look" like? Well, s is the linear functional that maps g_1 to $g_1(s)$, g_2 to $g_2(s)$, ... and g_k to $g_k(s)$. In other words, *metrized* s = $(g_1(s), g_2(s), \ldots g_k(s), \ldots)$. This last expression is nothing more or less than the result of the Construction application. But notice: deriving the result this way requires constructing the dual of Q, characterized as $\Sigma\lambda_k s_k$, $\lambda \in \mathbb{R}$, $s \in S$. In other words, **the expression $(\lambda_i s_i + \lambda_j s_j)$ now has meaning in a way that is consistent with the original topology τ defined on S.** The last statement above is the keystone for much that is to be developed below.

The point of all this discussion is that simple *algebraic operations* on the elements of S, namely vector addition and scalar multiplication, can be confidently done.

On the Plausibility of the Norm $||q|| = \int|q|$

A general line of attack to show that the metrics of S and Q are commensurable is to look for a norm on Q: a norm defined by the notion of the integral $\int|q|$ with respect to the topology τ on S. To firm up this notion, consider the following points:

- Do the elements of $Q = \{q: S \rightarrow R, q = \Sigma\lambda_k g_k\}$ have compact support: that is, do the elements of Q map to a non-zero value in R? Yes, because g_k is presumably continuous and open in some extension S' of S and some refinement τ' of τ; S' being some kind of *ultimate lexicon*.
- Is ε_s a positive Radon measure (a measure from utility theory)? Yes. Informally, one might consider any sequence of compact sets C_k where $\cap C_k = s$, where s is interior to C_k. The characteristic functions X_{Ck} converge weakly (in the dual):

$\varepsilon_s(q) = \lim_{k\to\infty} q(s)X_{Ck}(s)$. The linear form ε_s is often called the *Dirac measure* at the point s. Note that we have implicitly adopted the premise that S is locally compact.

Given a positive Radon measure μ on S, μ can be extended to the upper integral μ^* for positive functions on S. This leads to the definition of a semi-norm for functions on S, which in turn leads to the space $\mathcal{L}^1(S, \mu)$ (by completing Q with respect to the semi-norm). The norm on $\mathcal{L}^1(S, \mu)$ then reflects back (via duality) into S as $||s|| = \lim \int |q\, X_{Ck}|$.

Note that if Q is convex, then S spans a set that sits on the convex hull of Q, just as one would expect that the so-called "pure" states should.

The point of all this discussion is that simple algebraic operations on the elements of S that are *metric preserving* can now be confidently performed: namely vector addition and scalar multiplication.

On the Nature of the Elements of S

Consider the lexemes s_i = "mother" and s_j = "father." What is $(s_i + s_j)$? And in what sense is this sum compatible with the original topology τ?

$(s_i + s_j)$ is a vector that is very nearly co-linear with s_n = "parent," and indeed "parent" is an element (of the dual of Q) that is entailed by both "mother" and "father." One might say that s_n carries the potential to be instantiated as either s_i or s_j. Viewing the elements of S as *state vectors*, and adducing from this (and other examples), it becomes apparent that vector addition can be interpreted as corresponding to a *superposition* of states.

While the vector sum "mother" + "father" intuitively translates to the concept of "parent," other vector sums are less intuitively meaningful. Nevertheless, vector summation still operates to combine the vectors. What is "human" + "bird"? How about "turtle" + "electorate"? Even though these vector sums do not translate to a known concept in the dictionary, if the object is to combine the indicated vectors, superposition operates correctly.

Consider the (preliminary) proposition that **the sum of two state vectors corresponds to the superposition of the states of the addends**. If state vector addition corresponds to superposition of states, the question then naturally comes to mind, "What happens when we superpose a state with itself?" Ockham's razor suggests that the result of such an operation should yield the same state. From this we conjecture that if a state vector corresponding to a state is multiplied by any non-zero scalar, the resulting state vector

represents the same state. Put more succinctly, **semantic state is entailed in the direction of the state vector**.

Determining Semantic Abstracts

Now that superposition of state vectors has been shown to be feasible, one can construct semantic abstracts representing the content of the document as a vector within the topological vector space. FIG. 3 shows a two-dimensional topological vector space in which state vectors are used to determine a semantic abstract for a document. (FIG. 3 and FIG. 4 to follow, although accurate representations of a topological vector space, are greatly simplified for example purposes, since most topological vector spaces will have significantly higher dimensions.) In FIG. 3, the "x" symbols locate the heads of state vectors for terms in the document. (For clarity, the line segments from the origin of the topological vector space to the heads of the state vectors are eliminated.) Most of the state vectors for this document fall within a fairly narrow area of semantic content 305 in the topological vector space. Only a few outliers fall outside the core of semantic content 305.

The state vectors in semantic content 305 are superpositioned to form the semantic abstract. By taking the vector sum of the collected state vectors (the state vectors within semantic content 305), a single state vector 310 can be calculated as the semantic abstract.

Unit circle 315 marks all the points in the topological vector space that are a unit distance from the origin of the topological vector space. (In higher dimensional topological vector spaces, unit circle 315 becomes a unit hyper-sphere.) State vector 310 can be normalized to a unit distance (i.e., the intersection of state vector 310 and unit circle 315). Normalizing state vector 310 takes advantage of the (above-discussed) fact that semantic state is indicated by vector direction, and can compensate for the size of semantic content 305 used to construct state vector 310. One way to normalize state vector 310 is to divide the vector by its length: that is, if $\mathbf{v}$ is a state vector, $\mathbf{v}/\|\mathbf{v}\|$ is the unit vector in the direction of $\mathbf{v}$.

Measuring Distance between State Vectors

As discussed above, semantic state is entailed by the direction of the state vector. This makes sense, as the vector sum of a state with itself should still be the same state. It therefore makes the most sense to measure the distance between semantic abstract state vectors through the angle between the state vectors. In the preferred embodiment, distance is measured as the angle between the state vectors.

Distance can be measured as the distance between the heads of the state vectors. But recall that changing the length of two state vectors will change the distance between their heads. Since semantic state is entailed by the direction of the state vector, state vectors can be normalized without affecting their states before measuring distance as the difference of state vectors. Normalizing the state vectors allows a given distance between vectors to have a consistent meaning across different bases and state vectors.

FIG. 4 shows a two-dimensional topological vector space in which semantic abstracts for three documents are compared. In FIG. 4, three semantic abstracts represented as single state vectors 405, 410, and 415 are shown. Semantic abstract 405 (normalized from state vector 310 in FIG. 3) is the semantic abstract for the known document; semantic abstracts 410 and 415 are semantic abstracts for documents that may be similar to the document associated with semantic abstract 405. (Note that semantic abstracts 410 and 415 are also normalized.) Recall that distance can be measured as the angle between state vectors. The angle 420 between semantic abstracts 405 and 410 is relatively small, suggesting the two documents have similar content. In contrast, the angle 425 between semantic abstracts 405 and 415 is relatively large, suggesting the two documents have differing content.

Procedural Implementation

FIG. 5 is a flowchart of a method to determine a semantic abstract for a document in the system of FIG. 1. At step 505, the document's semantic content is determined. The semantic content of the document can be determined by using dominant vectors or dominant phrase vectors, as described in the Semantic Abstract application. (As further described in the Semantic Abstract application, after the vectors are constructed, they can be filtered to reduce the number of vectors factored into constructing the single vector for the semantic abstract.) At step 510, state vectors are constructed for each lexeme/lexeme phrase in the semantic content. At step 515, the state vectors are weighted, for example by multiplying the vectors with scaling factors. At step 520, the state vectors are superpositioned into a single vector using vector arithmetic. At step 525, the single vector is normalized. Finally, at step 530, the single vector is saved as the semantic abstract for the document.

Note that steps 515 and 525 are both optional. For example, the state vectors do not have to be weighted. Weighting the state vectors makes possible minimizing the weight of lexemes that are part of the semantic content but less significant to the document. And

normalizing the single vector, although highly recommended, is not required, since distance can be measured through angle.

The advantage of superpositioning the state vectors into a single vector is that the amount of storage required to store the semantic abstract. Whereas in the Semantic Abstract application, storing the semantic abstract requires storing several multi-dimensional state vectors, the invention only requires storing one multi-dimensional state vector. And, as shown above, because superpositioning state vectors does not lose information, storing the single state vector is as complete as storing the individual state vectors before superposition.

Having illustrated and described the principles of our invention in a preferred embodiment thereof, it should be readily apparent to those skilled in the art that the invention can be modified in arrangement and detail without departing from such principles. We claim all modifications coming within the spirit and scope of the accompanying claims.